

MC504/MC514 - Sistemas Operacionais

Processos e Threads: Introdução

Islene Calciolari Garcia

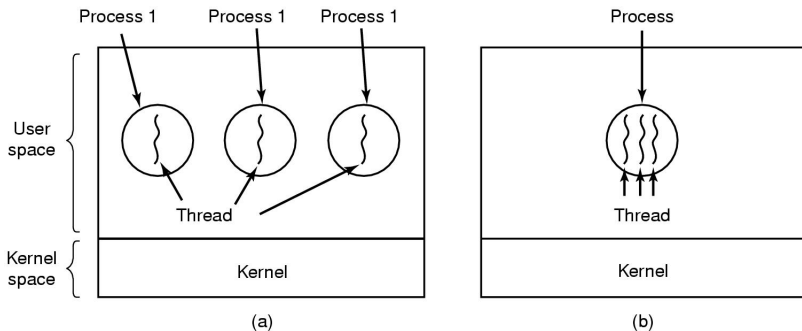
Segundo Semestre de 2014

Sumário

- 1 Espaço de endereçamento
- 2 Pthreads
 - create
 - join
 - Argumentos
 - exit
- 3 Pilhas de execução
- 4 Treinando

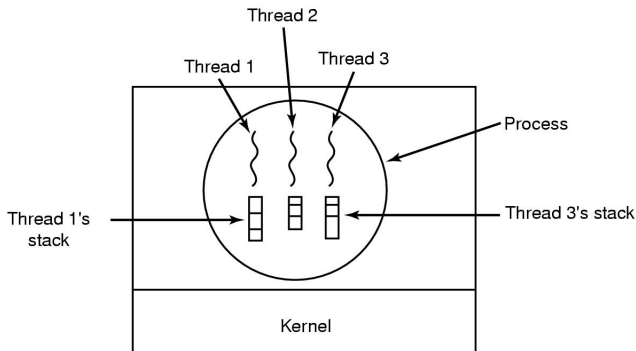
Threads

Várias threads podem compartilhar o mesmo espaço de endereçamento



Tanenbaum: Figura 2.6

Pilhas independentes



Tanenbaum: Figura 2.8

Como trabalhar com threads

Veja os comandos:

- `pthread_create`
- `pthread_join`
- `pthread_exit`

Para mais informações: `man <comando>`

create

Como criar uma thread

```
int  pthread_create(pthread_t  *thread,  
                    pthread_attr_t *attr,  
                    void * (*start_routine)(void *),  
                    void *arg);
```

Veja o código: create0.c

join

Como esperar por uma thread

```
int pthread_join(pthread_t thr,  
                 void **thread_return);
```

Veja os códigos: join0.c

Como passar argumentos para uma thread

- Exemplo: cada thread pode precisar de um identificador único.
- Veja os códigos: create1.c, create2.c, create3.c, create4.c e create5.c

exit

Como encerrar a execução de uma thread

- Comando `return` na função principal da thread (passada como parâmetro em `pthread_create`)
- Análogo ao comando `return` na função `main()`

Veja os códigos: `return0.c`, `return1.c` `pthread_return.c`

exit

Como encerrar a execução de uma thread

- `void pthread_exit(void *retval);`
- Análogo ao comando `exit(status);`

Veja os códigos: `exit0.c`, `exit1.c` e `pthread_exit0.c`

Vamos examinar a pilha de execução

- Componentes do frame (não necessariamente nesta ordem):
 - valor de retorno
 - endereço de retorno
 - registradores
 - argumentos para a função
 - variáveis locais da função
- No gdb execute comandos do tipo:

```
(gdb) p \x (int[10]) *s1
```

```
(gdb) p \x (int[10]) *(s1-2)
```

```
(gdb) set *(s1-2) = ...
```

Pilhas de execução

- Veja as pilhas de execução: pilhas.c
- Você acha que uma thread pode ...
 - ler o conteúdo da pilha de outra thread?
 - escrever na pilha de outra thread?
 - matar a outra thread? veja o código corrompe-pilhas.c

Treinando...

Você já pode escrever código multithread!

- Escreva códigos para soma, subtração e multiplicação de matrizes.
- Veja o laboratório proposto para a turma de 2010:
Pesca-palavras